The Voronoi Diagram of Convex Objects in the Plane

Menelaos Karavelas, Mariette Yvinec

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Voronoi Diagram of Convex Objects in the Plane

Menelaos Karavelas — Mariette Yvinec

N° 5023

December 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--5023--FR+ENG

The Voronoi Diagram of Convex Objects in the Plane

Menelaos Karavelas * , Mariette Yvinec †

Thème 2 — Génie logiciel
et calcul symbolique
Projet Geometrica

Rapport de recherche n° 5023 — December 2003 — 23 pages

Abstract: This paper presents a dynamic algorithm for the construction of the Euclidean Voronoi diagram of a set of convex objects in the plane. We consider first the Voronoi diagram of smooth convex objects forming pseudo-circles set. A pseudo-circles set is a set of bounded objects such that the boundaries of any two objects intersect at most twice. Our algorithm is a randomized dynamic algorithm. It does not use a conflict graph or any sophisticated data structure to perform conflict detection. This feature allows us to handle deletions in a relatively easy way. In the case where objects do not intersect, the randomized complexity of an insertion or deletion can be shown to be respectively $O(\log^2 n)$ and $O(\log^3 n)$. Our algorithm can easily be adapted to the case of pseudo-circles sets formed by piecewise smooth convex objects. Finally, given any set of convex objects in the plane, we show how to compute the restriction of the Voronoi diagram in the complement of the objects' union.

Key-words: Vorono diagram; Delaunay triangulation; Euclidean distance; abstract Voronoi diagram; randomized algorithm; dynamic algorithms

Work partially supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-2000-26473 (ECG - Effective Computational Geometry for Curves and Surfaces).

A preliminary version of this paper has appeared in: *The Voronoi Diagram of Planar Convex Objects*, Proceedings of the 11th Annual European Symposium on Algorithms, LNCS 2832, pp. 337-348, Hungary, 2003.

* Computer Science & Engineering Department, University of Notre Dame, Notre Dame, IN 46556, U.S.A.; email: mkaravel@cse.nd.edu.

† INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France; email: Mariette.Yvinec@sophia.inria.fr.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Diagrammes de Voronoï d'Objets Convexes en Dimension 2

Résumé : Ce rapport présente un algorithme dynamique pour construire le diagramme de Voronoi Euclidien d'un ensemble d'objets convexes en dimension 2. Nous considérons tout d'abord la cas d'objets lisses et convexes constituant un ensemble de pseudo-circles. Un ensemble de pseudo-circles est un ensemble d'objets bornés tels que les frontières de deux objets quelconques de l'ensemble ont au plus deux points d'intersection. Notre algorithme est randomisé et dynamique. Il n'utilise ni graphes de conflits ni structures sophistiquée pour détecter les conflits. De ce fait, il est relativement facile de gérer les suppressions. Si les objets sont disjoints, le coût randommisé d'une insertion et $O(\log^2 n)$ et celui d'une suppresssion est $O(\log^3 n)$ Cet algorithme peut être adaptée aux ensembles de pseudo-circles formés d'objets convexes lisses par morceaux. Pour finir, il permet aussi de calculer, pour tout ensemble d'objets convexes, la restriction de leur diagramme de Voronoï au complémentaire de leur union.

Mots-clés : diagramme de Voronoï; triangulation de Delaunay; distance euclidienne; diagramme de Voronoï abstrait; algorithme randomisés, algorithme dynamique

1 Introduction

Given a set of sites and a distance function from a point to a site, a Voronoi diagram can be roughly described as the partition of the space into cells that are the locus of points closer to a given site than to any other site. Voronoi diagrams have proven to be useful structures in various fields such as astronomy, crystallography, biology etc. Voronoi diagrams have been extensively studied. See for example the survey by Aurenhammer [2] or the more recent one by Aurenhammer and Klein [3] or the book by Okabe, Boots, Sugihara and Chiu [12]. The early studies were mainly concerned with point sites and the Euclidean distance. Subsequent studies considered extended sites such has segments, lines, convex polytopes and various distances such as L_1 or L_∞ or any distance defined by a convex polytope as unit ball. While the complexity and the related algorithmic issues of Voronoi diagrams for extended sites in higher dimensions is still not completely understood, as witnessed in the recent works by Koltun and Sharir [9, 10], the planar cases are now rather well mastered, at least for linear objects. The rising need for handling curved objects triggered further works for the planar cases. Klein et al. [7, 8] set up a general framework of *abstract Voronoi diagrams* which covers a large class of planar Voronoi diagrams. They provided a randomized incremental algorithm to construct diagrams of this class. Alt and Schwarzkopf [1] handled the case of generic planar curves and described an incremental randomized algorithm for this case too. Since they handle curves, they cannot handle objects with non-empty interior, which is our focus. Their algorithm is incremental but does not work in-line (it requires the construction of a Delaunay triangulation with one point on each curve before the curve segments are really treated). Another closely related work is that by McAllister, Kirkpatrick and Snoeyink [11], which deals with the Voronoi diagrams of disjoint convex polygons. The algorithm presented treats the convex polygons as objects, rather than as collections of segments; it follows the sweep-line paradigm, thus it is not dynamic. Moreover, the case of intersecting convex polygons is not considered. The present papers deals with the Euclidean Voronoi diagram of planar convex objects and generalizes a previous work of the same authors on the Voronoi diagram of circles [6].

Let p be a point and A be a bounded convex object in the Euclidean plane $\mathbb{E}^2$. We define the distance $\delta(p, A)$ from p to A to be:

$$\delta(p, A) = \begin{cases} \min_{x \in \partial A} \|p - x\|, & x \notin A \\ -\min_{x \in \partial A} \|p - x\|, & x \in A \end{cases}$$

where ∂A denotes the boundary of A and $\|\cdot\|$ denotes the Euclidean norm.

Given the distance $\delta(\cdot, \cdot)$ and a set of convex objects $\mathcal{A} = \{A_1, \ldots, A_n\}$, the *Voronoi diagram* $\mathcal{V}(\mathcal{A})$ is the planar partition into cells, edges and vertices defined as follows. The Voronoi cell of an object A_i is the set of points which are closer to A_i than to any other object in $\mathcal{A}$. Voronoi edges are maximal connected sets of points equidistant to two objects in $\mathcal{A}$ and closer to these objects than to any other in $\mathcal{A}$. Voronoi vertices are points equidistant to at least three objects of $\mathcal{A}$ and closer to these objects than to any other object in $\mathcal{A}$.

We first consider Voronoi diagrams for special collections of smooth convex objects called *pseudo-circles sets*. A pseudo-circles set is a set of bounded objects such that the boundaries of any two objects in the set have at most two intersection points. In the sequel, unless specified oth-

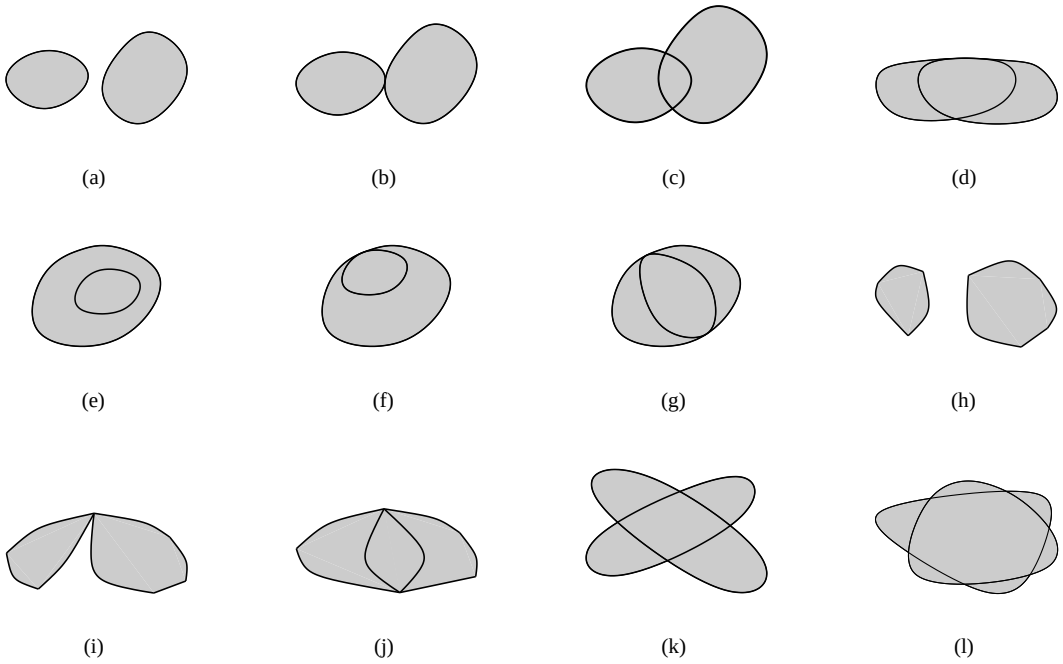

Figure 1: Various configurations for two convex objects. (a)-(e): an sc-pseudo-circles set in general position; (f)-(g): an sc-pseudo-circles set in degenerate position; (h)-(j): a pseudo-circles set of piecewise smooth convex objects; (k)-(l): not a pseudo-circles set.

erwise, we consider pseudo-circles sets formed by smooth convex objects, and we call them *smooth convex pseudo-circles sets*, or *sc-pseudo-circles sets* for short.

Let A be a convex object. A line L is a *supporting line* of A if and only if A is included in one of the closed half-planes bounded by L, and $\partial A \cap L$ is not empty. Given two convex objects A_i and A_j, a line L is a *(common) supporting line* of A_i and A_j if and only if L is a supporting line of A_i and A_j, such that A_i and A_j are both included in the same half-plane bounded by L.

In this paper, we first deal with smooth bounded convex objects forming pseudo-circles sets (cf. Fig. 1). Any two objects in such a set set have at most two common supporting lines (cf. Fig. 1). Two convex objects have no common supporting line if one is included in the other. They have two common supporting lines if they are either disjoint or properly intersecting at two points (a proper intersection point is a point where the boundaries are not only meeting but also crossing each other) or externally tangent (which means that their interiors are disjoint and their boundaries share a common tangent point). Two objects forming a pseudo-circles set may also be internally tangent, meaning that one is included in the other and their boundaries share one or two common points.

Then they have, respectively, one or two common supporting lines. A pseudo-circles set is said to be in *general position* if there is no pair of tangent objects. In fact, tangent objects which are properly intersecting at their common tangent point or externally tangent objects do not harm our algorithm and we shall say that a pseudo-circles set is in general position when there is no pair of internally tangent objects.

The algorithm that we propose for the construction of the Voronoi diagram of sc-pseudo-circles sets in general position is a dynamic one. It is a variant of the incremental randomized algorithm proposed by Klein et al. [8]. The data structures used are simple, which allows us perform not only insertions but also deletions of sites in a relatively easy way. When input sites are allowed to intersect each other, it is possible for a site to have an empty Voronoi cell. Such a site is called a *hidden* site, while a site with non-empty cell is said to be *visible*. Our algorithm handles hidden sites. The detection of the first conflict or the detection of a hidden site is performed through closest site queries. Such a query can be done by either a simple walk in the Voronoi diagram or using a hierarchy of Voronoi diagrams, i.e., a data structure inspired from the Delaunay hierarchy of Devillers [5].

To analyze the complexity of the algorithm, we assume that each object has constant complexity, which implies that each operation involving a constant number of object is performed in constant time (e.g., finding a circle tangent to three objects). We show that if sites do not intersect, the randomized complexity of updating a Voronoi diagram with n sites is $O(\log^2 n)$ for an insertion and $O(\log^3 n)$ for a deletion. The complexities of insertions and deletions are more involved when sites intersect.

We then extend our results by firstly dropping the hypothesis of general position and secondly by dealing with pseudo-circles sets formed by convex objects whose boundaries are only piecewise smooth. Using this extension, we can then build the Voronoi diagram of any set $\mathcal{A}$ of convex objects in the complement of the objects' union (i.e., in free space). This done by constructing a new set of objects $\mathcal{A}'$, which is a pseudo-circles set of piecewise smooth convex objects and such that the Voronoi diagrams $\mathcal{V}(\mathcal{A})$ and $\mathcal{V}(\mathcal{A}')$ coincide in free space.

The rest of the paper is structured as follows. In Section 2 we study the properties of the Euclidean Voronoi diagram of sc-pseudo-circles sets in general position. In particular we show that such a diagram belongs to the class of abstract Voronoi diagrams described by Klein et al. [8]. In Section 3 we present the dynamic algorithm for the construction of the Voronoi diagram of sc-pseudo-circles sets in general position. Section 4 describes closest site queries, whereas Section 5 deals with the complexity analysis of insertions and deletions. Finally, in Section 6 we show how our approach can be extended to handle sc-pseudo-circles sets with degeneracies, pseudo-circles sets of convex objects with piecewise smooth boundaries and eventually any set of convex objects in the plane.

2 The Voronoi diagram of sc-pseudo-circles sets

In this section we present the main properties of the Voronoi diagram of sc-pseudo-circles sets in general position. Let us first make precise a few definitions and notations. Here and in the following, we consider any bounded convex object A_i as closed and we note ∂A_i and A_i°, respectively, the boundary and the interior of A_i.

Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be an sc-pseudo-circles set. The Voronoi cell of an object A is denoted as $V(A)$ and is considered a closed set. The interior and boundary of $V(A)$ are denoted by $V^\circ(A)$ and $\partial V(A)$, respectively. We are going to consider maximal disks either included in a given object A_i or disjoint from A_i°, where the term maximal refers to the inclusion relation. For any point x, we denote by $C_i(x)$ the closed disk centered at x with radius $|\delta(x, A_i)|$. If $x \notin A_i$, $C_i(x)$ is the maximal disk centered at x and disjoint from A_i°. If $x \in A_i$, $C_i(x)$ is the maximal disk centered at x and included in A_i. In the latter case these is a unique maximal disk inside A_i containing $C_i(x)$, which we denote by $M_i(x)$. Finally, the *medial axis* $S(A_i)$ of a bounded convex object A_i is defined as the locus of points that are centers of maximal disks included in A_i.

Let A_i and A_j be two smooth bounded convex objects. The set of points $p \in \mathbb{E}^2$ that are at equal distance from A_i and A_j is called the bisector π_{ij} of A_i and A_j. Theorem 2 ensures that π_{ij} is an one-dimensional set if the two objects A_i and A_j form an sc-pseudo-circles set in general position and justifies the definition of Voronoi edges given above.

Let us begin with a technical lemma.

Lemma 1 *Let A_i and A_j be two bounded convex objects.*

1. *If $x \in A_i$, $\delta(x, A_i) < \delta(x, A_j)$ if and only if $C_i(x)$ is not included in A_j and $\delta(x, A_i) = \delta(x, A_j)$ if and only if $C_i(x)$ is internally tangent to A_j.*
2. *If $x \notin A_i$, $\delta(x, A_i) < \delta(x, A_j)$ if and only if $C_i(x)$ does not intersect A_j, and $\delta(x, A_i) = \delta(x, A_j)$ if and only if $C_i(x)$ is externally tangent to A_j.*

Proof. Follows trivially from the definition of the distance. □

Theorem 2 *Let A_i and A_j be two convex objects forming a pseudo-circles set in general position and let π_{ij} be the bisector of A_i, A_j with respect to the Euclidean distance $\delta(\cdot, \cdot)$. Then :*

1. *If A_i and A_j have no supporting line, then $\pi_{ij} = \emptyset$.*
2. *If A_i and A_j have two supporting lines, then π_{ij} is a single curve homeomorphic to the open interval $(0, 1)$.*

Proof.

1. Suppose that A_i and A_j have no common supporting line. This implies that either $A_i \subset A_j^\circ$ or $A_j \subset A_i^\circ$. Let us assume that $A_j \subset A_i^\circ$. Let $x \in \mathbb{E}^2$. We consider the following cases for x :
   (a) $x \notin A_i$. Any disk centered in x that does not intersect A_i° does not intersect A_j. This is in particular true for $C_i(x)$ which implies that $\delta(x, A_i) < \delta(x, A_j)$.
   (b) $x \in A_i \setminus A_j$. Then $\delta(p, A_i) \leq 0 < \delta(p, A_j)$.
   (c) $x \in A_j$. The maximum disk $C_i(x)$ is tangent to ∂A_i at at least one point and therefore cannot be included in A_j. Thus, $\delta(p, A_i) < \delta(p, A_j)$.

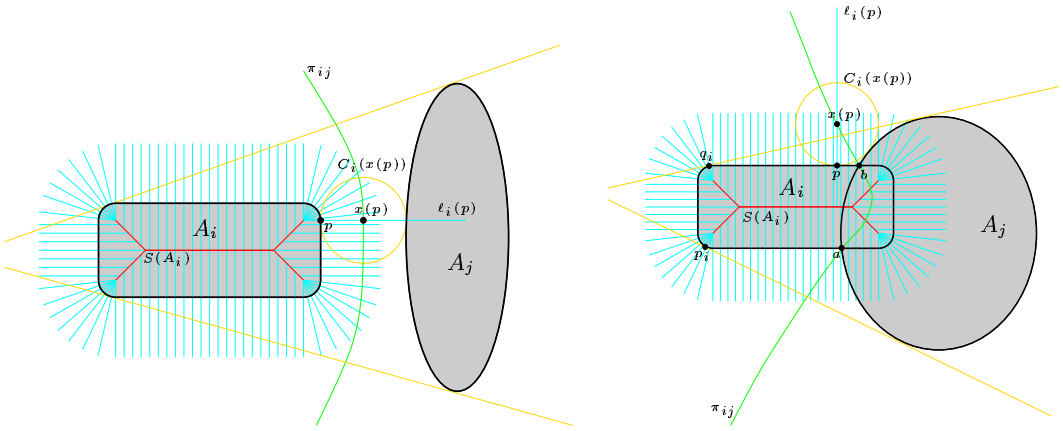

Figure 2: The set of rays $\ell_i(y)$, $y \in \partial A_i$, covers the entire plane. The ray $\ell_i(p)$ and the bisector π_{ij} intersect in at most one point. Left: the case $A_i \cap A_j = \emptyset$. Right: the case $A_i \cap A_j \neq \emptyset$.

Thus any point x is closer to A_i than A_j, and $\pi_{ij} = \emptyset$.

2. Suppose now that A_i, A_j have two common supporting lines. For a point $p \in \partial A_i$, let $\sigma_i(p)$ be the point of the medial axis $S(A_i)$ which is the center of the maximal disk included in A_i and tangent to ∂A_i at p. We denote by $\ell_i(p)$ the half-line issued from $\sigma_i(p)$ and perpendicular to ∂A_i at p. The set of rays $\ell_i(p)$ covers the whole plane and two such rays do not intersect except if they share the same origin on the medial axis. (See Fig. 2). We first show that the bisector π_{ij} intersects each ray $\ell_i(p)$ in at most one point.

   For any point x on $\ell_i(p)$, we consider as before the disk $C_i(x)$ centered at x with radius $|\delta(x, A_i)|$. $C_i(x)$ is tangent to ∂A_i at p. When x moves from $\sigma_i(p)$ to p along $\ell_i(p)$, $C_i(x)$ decreases from the maximal disk $M_i(\sigma(p))$ to the disk $C_i(p)$ which is reduced to the point p. Then, when x moves beyond p on $\ell_i(p)$, the disk $C_i(x)$ increase from p to the closed halfplane $H_i(p)$ limited by the line tangent to ∂A_i at p and not containing A_i.

   We denote by p_i, q_i the contact points of object A_i with the common supporting lines of A_i and A_j, and p_j, q_j the contact points of object A_j with those lines. We assume that the contact points p_i, q_i, p_j and q_j are labeled in such a way that they are encountered in that order along a counterclockwise traversal of the boundary of the convex hull $CH(A_i \cup A_j)$. Let α_i be the open arc between q_i and p_i along a counterclockwise traversal of ∂A_i, i.e. α_i is the part of ∂A_i which does not appear on $\partial CH(A_i \cup A_j)$.

   We will first assume that the contact points of A_i and A_j with their supporting lines are distinct, i.e., $p_i \neq q_j$ and $q_i \neq p_j$.

   If $p \notin \alpha_i$, then first of all $p \notin A_j$ (this would contradict the fact that p lies on $\partial CH(A_i \cup A_j)$). We then claim that any point $x \in \ell_i(p)$ is closer to A_i and $\ell_i(p) \cap \pi_{ij} = \emptyset$. Indeed, for any

point $x \in \sigma_i(p)p$, the disk $C_i(x)$ includes p and thus it is not included in A_j and for any point x on $\ell_i(p)$ beyond p, $C_i(x)$ is included in the halfplane $H_i(p)$ which does not intersect A_j.

If $p \in \alpha_i$ but $p \notin A_j$, π_{ij} intersects $\ell_i(p)$ at a single point. Indeed, any point $x \in \sigma_i(p)p$ is closer to A_i, for the same reason as above. Let us consider now the points of $\ell_i(p)$ beyond p. When x moves beyond p, $C_i(x)$ increases from $C_i(p) = \{p\}$ which does not intersect A_j to the halfplane $H_i(p)$ which does. There is a unique point $x(p)$ beyond p on $\ell_i(p)$ for which $C_i(x(p))$ is externally tangent to A_j. This point is at equal distance from A_i and A_j, and thus belongs to π_{ij}. Note that the uniqueness of $x(p)$ stems from the fact that $C_i(y) \subseteq C_i(y')$ if $\|y - p\| \leq \|y' - p\|$.

At last, when A_i and A_j intersect there are points p in $\alpha_i \cap A_j$. For such a point p, $\ell_i(p) \cap \pi_{ij}$ can be either empty or a single point. Indeed any point x beyond p on $\ell_i(p)$ is closer to A_j, because the disk $C_i(x)$ includes p and thus intersects A_j. Consider now points x on $\sigma_i(p)p$. Assume first that $M_i(\sigma_i(p))$ is included in A_j. Then, for any $x \in \sigma_i(p)p$, $C_i(x)$ is included in A_j, x is closer to A_j and $\ell_i(p) \cap \pi_{ij}$ is empty. Assume now that $M_i(\sigma_i(p))$ is not included in A_j. When x moves from $\sigma_i(p)$ to p, $C_i(x)$ decreases from $M_i(\sigma_i(p))$, which is not included in A_j, to $\{p\}$ which is included in A_j. There exists a unique point $x(p) \in \sigma_i(p)p$ such that $C_i(x(p))$ is tangent to ∂A_j. This point is at equal distance from A_i and A_j and thus belongs to π_{ij}. Again we can argue the uniqueness of $x(p)$ using an arguments similar to the one above.

Thus if A_i and A_j are disjoint, then for any point $p \in \alpha_i$, there is a unique point $x(p)$ in $\ell_i(p) \cap \pi_{ij}$. Reciprocally, any point y in π_{ij} is the center of a disk tangent to both A_i and A_j. This disk touches ∂A_i in a point p of α_i such that $y = x(p)$. Thus the mapping from α_i to π_{ij}, which maps $p \in \alpha_i$ to the point $x(p) = \ell_i(p) \cap \pi_{ij}$ is an one-to-one and onto mapping. The reverse mapping is well known to be continuous and therefore π_{ij} is a single curve homeomorphic to the open arc α_i, i.e., to the open interval $(0, 1)$.

Assume now that A_i and A_j are not disjoint. From the pseudo-circles property ∂A_i and ∂A_j intersect in two points a and b. Assume that q_i, b, a and p_i are encountered in that order on the counterclockwise traversal of the arc α_i. Let α_i^1 be the subarc of α_i joining q_i to b. Any point p on α_i^1 is on $\alpha_i \setminus A_j$ and therefore the ray $\ell_i(p)$ intersects the bisector π_{ij} at a unique point $x(p)$. As above the mapping from p to $x(p)$ is one-to-one and continuous and therefore the bisector π_{ij} includes an unbounded simple arc π_{ij}^1 homeomorphic to α_i^1, (i.e., homeomorphic to the interval $(0, 1]$), and joining $b = \pi_{ij}(b)$ with the image of q_i which is the infinite point of the ray $\ell_i(q_i)$. In the same way, if α_i^3 is the subarc of α_i joining a to p_i, the bisector π_{ij} includes an unbounded simple arc π_{ij}^3 homeomorphic to α_i^3 (i.e., homeomorphic to the interval $[0, 1)$), and joining $a = \pi_{ij}(a)$ to the image of p_i which is the infinite point of the ray $\ell_i(p_i)$. These two arcs are disjoint, they are included in the complement of $A_i \cup A_j$ and are the only components of the bisector in that region. Obviously π_{ij} has no component in $A_i \setminus A_j$ and $A_j \setminus A_i$. Let us show now that $\pi_{ij} \cap A_i \cap A_j$ is a simple connected curve joining a to b. First let us notice that π_{ij} has to include a connected component in $A_i \cap A_j$ joining a to b. Indeed consider the continuous function $f(x) = \delta(x, A_i) - \delta(x, A_j)$. Let x_i be a point of $\partial A_i \cap A_j$ and x_j a point of $\partial A_j \cap A_i$. Assume that x_i and x_j are distinct from a and b. Then we have $f(x_i) > 0$, $f(x_j) < 0$. Thus there exists at least one point where $f(x) = 0$, i.e. a point of

π_{ij} on any path joining x_i to x_j in $A_i \cap A_j$. Then we remark that any point in $\pi_{ij} \cap A_i \cap A_j$ belongs to the medial axis of the convex body $A_i \cap A_j$. The medial axis of this object is a tree and has a single path joining a to b. This proves that $\pi_{ij} \cap A_i \cap A_j$ is the path of the medial axis joining a to b. Finally the concatenation of the three arcs π_{ij}^1, π_{ij}^3 and $\pi_{ij}^2 = \pi_{ij} \cap A_i \cap A_j$ yields a curve homeomorphic to $(0, 1)$.

To finish the proof we need to consider the case where the points of contact of the supporting lines coincide. For convenience we will assume that both $p_i \equiv q_j$ and $q_i \equiv p_j$. The remaining cases are just combinations of what we describe below and the arguments made above.

Note that in this case the common points of contact are also the points of intersection of the boundaries ∂A_i and ∂A_j. Moreover, the arc α_i^1 (resp. α_i^3) is now a ray starting from b (resp. a), with direction perpendicular to ∂A_i and ∂A_j, that is contained in the closed halfspace $H_i(b)$ (resp. $H_i(a)$). As far as the portion of the bisector inside $A_i \cap A_j$ is concerned we can no longer claim that is it the portion of the medial axis $S(A_i \cap A_j)$ connecting b with a; this is due to the fact that the points a and b are no longer points of discontinuity on the boundary $\partial(A_i \cap A_j)$ of $A_i \cap A_j$ and thus they are not necessarily points on $S(A_i \cap A_j)$. However, the same argument works with minor modifications. Let a' and b' be the points on the medial axis $S(A_i \cap A_j)$ corresponding to a and b, respectively. Clearly, there is a unique path Π from a' to b' in $S(A_i \cap A_j)$ (recall that the medial axis $S(A_i \cap A_j)$ is a tree, since $A_i \cap A_j$ is a convex object). Now consider the path $bb' \cup \Pi \cup a'a$. This path connects b to a and all its points are at equal distance from the two arcs $\partial A_i \cap A_j$ and $\partial A_j \cap A_i$ on the boundary of $A_i \cap A_j$. The union of this path along with the two rays emanating from a and b constitute the bisector π_{ij} of A_i and A_j. Clearly, π_{ij} is homeomorphic to the interval $(0, 1)$.

□

Theorem 4 ensures that each cell in the Euclidean Voronoi diagram of an sc-pseudo-circles set in general position is simply connected. We begin by a technical lemma which generalizes Lemma 1.

Lemma 3 *Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be an sc-pseudo-circles set.*

1. *If $x \in A_i$, then x belongs to the Voronoi cell $V(A_i)$ of A_i if and only if $C_i(x)$ is not contained in the interior A_j° of any object A_j in $\mathcal{A} \setminus \{A_i\}$.*
2. *If $x \in A_i$, then x belongs to the interior $V^\circ(A_i)$ of the Voronoi cell $V(A_i)$ of A_i if and only if $C_i(x)$ is not contained in any object A_j in $\mathcal{A} \setminus \{A_i\}$.*
3. *If $x \notin A_i$, then x belongs to the Voronoi cell $V(A_i)$ of A_i if and only if $C_i(x)$ does not intersect the interior A_j° of any object A_j in $\mathcal{A} \setminus \{A_i\}$.*
4. *If $x \notin A_i$, then x belongs to the interior $V^\circ(A_i)$ of the Voronoi cell $V(A_i)$ of A_i if and only if $C_i(x)$ does not intersect any object A_j in $\mathcal{A} \setminus \{A_i\}$.*

Proof. The proof follows trivially from Lemma 1. □

Theorem 4 *Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be an sc-pseudo-circles set in general position. For each object A_i, we denote by $N(A_i)$ the locus of the centers of maximal disks included in A_i that are not included in the interior of any object in $\mathcal{A} \setminus \{A_i\}$, and by $N^\circ(A_i)$ the locus of the centers of maximal disks included in A_i that are not included in any object in $\mathcal{A} \setminus \{A_i\}$. Then:*

1. *$N(A_i) = S(A_i) \cap V(A_i)$ and $N^\circ(A_i) = S(A_i) \cap V^\circ(A_i)$.*
2. *$N(A_i)$ and $N^\circ(A_i)$ are simply connected sets.*
3. *The Voronoi cell $V(A_i)$ is weakly star-shaped with respect to $N(A_i)$, which means that any point of $V(A_i)$ can be connected to a point in $N(A_i)$ by a segment included in $V(A_i)$. Analogously, $V^\circ(A_i)$ is weakly star-shaped with respect to $N^\circ(A_i)$.*
4. *$V(A_i) = \emptyset$ if and only if $N(A_i) = \emptyset$ and $V^\circ(A_i) = \emptyset$ if and only if $N^\circ(A_i) = \emptyset$.*

Proof. For any point $x \in \mathbb{E}^2$, we note as before $C_i(x)$ the disk centered at x with radius $|\delta(x, A_i)|$ and by $p_i(x)$ the point where $C_i(x)$ touches ∂A_i. If $x \in A_i$ the disk $C_i(x)$ is included in a unique maximal disk inside A_i which is called $M_i(x)$. If $x \notin A_i$, we still denote by $M_i(x)$ the maximal disk included in A_i and tangent to ∂A_i at $p_i(x)$. In any case, we note $\sigma_i(x)$ the center of $M_i(x)$.

1. Let $y \in N(A_i)$. By definition $y \in S(A_i)$. Consider the circle $C_i(y) = M_i(y)$. Since it is not contained in the interior A_j° of any object in $\mathcal{A} \setminus \{A_i\}$, we have by Lemma 3, Case 2, that $y \in V(A_i)$. Conversely, if $y \in S(A_i) \cap V(A_i)$, then we have by Lemma 3, Case 2, that $M_i(y)$ is not contained in the interior of some other object A_j in $\mathcal{A}$. Hence $y \in N(A_i)$. The proof of the other assertion is analog.

2. Let u' and v' be two points in $N(A_i)$. Because $S(A_i)$ is a tree, there is a unique path $\mathcal{P}$ in $S(A_i)$ connecting u' to v'. Suppose that there exists a point $w \in \mathcal{P}$ such that $w \notin N(A_i)$. This implies that $M_i(w)$ is contained in the interior A_j° of some other object in $\mathcal{A}$. Consider the subpath of $\mathcal{P}$ from w to u'. Since $M_i(w)$ is contained in A_j° while $M_i(u')$ is not, there must be a first point u on this subpath (from w to u') such that $M_i(u)$ is tangent to A_j. Similarly, there exists a first point v in the subpath of $\mathcal{P}$ from w to v' such that $M_i(v)$ is tangent to A_j. The end of the proof amounts to show that this situation enforces the existence of more than two intersection points between ∂A_i and ∂A_j, which contradicts with the fact A_i and A_j belong to a pseudo-circles set.

   Let p_u and q_u (resp. p_v and q_v) be the contact points between $M_i(u)$ (resp $M_i(v)$) and ∂A_i and let r_u (resp. r_v) be the contact point of $M_i(u)$ (resp $M_i(v)$) with ∂A_j. (See Fig. 3). Assume p_u, q_u, p_v, q_v are labeled in such a way that they are encountered in that order along a counterclockwise traversal of the boundary of the convex hull $CH(M_i(u) \cup M_i(v))$. Because any maximal ball in A_i centered between u and v on $\mathcal{P}$ is included in A_j°, r_u (resp. r_v) is encountered between p_u and q_u (resp. between p_v and q_v) on $\partial CH(M_i(u) \cup M_i(v))$. Let γ be the simple closed path which counterclockwisely follows $\partial CH(M_i(u) \cup M_i(v))$ from r_u to r_v and ∂A_j from r_v to r_u. Except if A_i and A_j are internally tangent, which contradicts the general position assumption, r_u and r_v are in A_i° and on ∂A_j while p_u, q_u, p_v, q_v are in

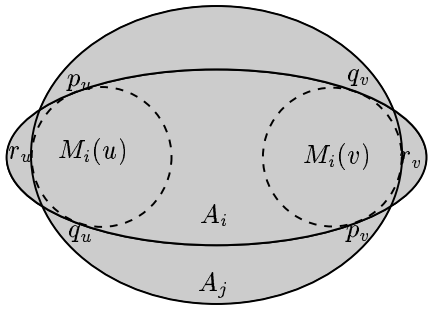

Figure 3: The boundaries of A_i and A_j have at least four intersection points.

A_j° and on ∂A_i. Thus γ which passes through r_u, q_u, p_v, r_v and encloses q_v and p_u has to intersect ∂A_i at least twice. This implies that the arc of ∂A_j joining counterclockwisely r_v to r_u intersects twice ∂A_i. In the same way the arc of ∂A_j joining counterclockwisely r_u to r_v intersects twice ∂A_i. Since the two arcs on ∂A_j are disjoint (except for r_u and r_v, of course), we must have at least four intersection points between ∂A_j and ∂A_i, which contradicts the fact that $\mathcal{A}$ is a pseudo-circle sets.

3. Consider a point $x \in V(A_i)$. We will show that $\sigma_i(x) \in N(A_i)$ and that any point y of the segment $x\sigma_i(x)$ belongs to $V(A_i)$. If $x \in A_i$, we grow a circle tangent to ∂A_i at $p_i(x)$ from $C_i(x)$ to $M_i(x)$. Any point y in $x\sigma_i(x)$ is the center of a circle $C_i(y)$ tangent to ∂A_i at $p_i(x)$ and such that $C_i(x) \subseteq C_i(y) \subseteq M_i(x)$. Because $x \in V(A_i)$, $C_i(x)$ is not contained in the interior A_j° of any other object of $\mathcal{A}$, hence $C_i(y)$ and $M_i(x)$ are not included in the interior A_j° of any other object A_j, which proves that $y \in V(A_i)$ and $\sigma_i(x) \in N(A_i)$. If $x \notin A_i$, we first shrink a circle tangent to ∂A_i at $p_i(x)$, from $C_i(x)$ to the point $p_i(x)$, then grow a circle tangent to ∂A_i at $p_i(x)$ from the point $p_i(x)$ to $M_i(x)$. Any point y in the subsegment $xp_i(x)$ is the center of a circle $C_i(y)$ tangent to ∂A_i at $p_i(x)$ and such that $C_i(y) \subseteq C_i(x)$. Because $x \in V(A_i)$, $C_i(x)$ does not intersect the interior A_j° of any other object of $\mathcal{A}$, thus neither does $C_i(y)$ and $y \in V(A_i)$. Any point y in subsegment $p_i(x)\sigma_i(x)$ is the center of a circle $C_i(y)$ tangent to ∂A_i at $p_i(x)$ and such that $p_i(x) \subseteq C_i(y) \subseteq M_i(x)$. Because $p_i(x) \in V(A_i)$, $p_i(x)$ is not contained in the interior A_j° of any other object A_j of $\mathcal{A}$, and thus neither does $C_i(y)$ nor $M_i(x)$. Thus $y \in V(A_i)$ and $\sigma_i(x) \in N(A_i)$.

   We can apply an analogous argument to show that if $x \in V^\circ(A_i)$, then any point y in $x\sigma_i(x)$ belongs to $V^\circ(A_i)$.

4. Claim 4 is follows immediately from the proofs of the previous claims.

□

In the sequel we say that an object A is *hidden* if $N^\circ(A) = \emptyset$.

In the framework of abstract Voronoi diagrams introduced by Klein [7], the diagram is defined by a set of bisecting curves $B_{i,j}$. In this framework, a set of bisectors is set to be *admissible* if:

1. Each bisector is homeomorphic to a line.
2. The closures of the Voronoi regions covers the entire plane.
3. Regions are path connected.
4. Two bisectors intersect in at most a finite number of connected components.

Let us consider the Euclidean Voronoi diagrams of an sc-pseudo-circles set in general position such that any pair of objects has exactly two supporting lines. Theorems 2 and 4 ensure, respectively, that Conditions 1 and 3 are fulfilled. Condition 2 is granted for any diagram induced by a distance. Condition 4 is a technical condition that we have not explicitly proved. In our case this results indeed from the assumption that the objects have constant complexity (note that Condition 4 is used in the theory of abstract Voronoi diagrams to prove that Voronoi cells are simply connected, which in our case is directly ensured by Theorem 4). The converse is also true: if we have a set of convex objects in general position, then their bisectors form an admissible system only if every pair of objects has exactly two supporting lines. Indeed, if this is not the case, one of the following holds: (1) the bisector is empty (e.g., if one object is contained in the interior of another; cf. Fig. 1(e)); (2) there exist Voronoi cells that consist of more than one connected components (e.g., if two the boundaries of two objects have four points of intersection; cf. Fig. 1(k)).

Theorem 5 *Let $A = \{A_1, \ldots, A_n\}$ be a set of smooth convex objects of constant complexity and in general position. Then the set of bisectors π_{ij} is an admissible system of bisectors if and only if every pair of objects has exactly two supporting lines.*

3 The dynamic algorithm

The algorithm that we propose is a variant of the randomized incremental algorithm for abstract Voronoi diagrams proposed by Klein and al. [8]. Our algorithm is fully dynamic and maintains the Voronoi diagram when a site is either added to the current set or deleted from it. To facilitate the presentation of the algorithm we first define the compactified version of the diagram and introduce the notion of conflict region.

The compactified diagram. We call 1-skeleton of the Voronoi diagram, the union of the Voronoi vertices and Voronoi edges. The 1-skeleton of the Voronoi diagram of an sc-pseudo-circles set $\mathcal{A}$ may consist of more than one connected components. However, we can define a compactified version of the diagram by adding to $\mathcal{A}$ a spurious site, A_∞ called the infinite site. The bisector of A_∞ and $A_i \in \mathcal{A}$ is a closed curve at infinity, intersecting any unbounded edge of the original diagram (see for example [7]). In the sequel we consider such a compactified version of the diagram, in which case the 1-skeleton is connected.

The conflict region. Each point x on a Voronoi edge incident to $V(A_i)$ and $V(A_j)$ is the center of a disk $C_{ij}(x)$ tangent to the boundaries ∂A_i and ∂A_j. This disk is called a *Voronoi bitangent disk*, and

more precisely an interior Voronoi bitangent disk if it is included in $A_i \cap A_j$, or an exterior Voronoi bitangent disk if it lies in the complement of $A_i^\circ \cup A_j^\circ$. Similarly, a Voronoi vertex that belongs to the cells $V(A_i)$, $V(A_j)$ and $V(A_k)$ is the center of a disk $C_{ijk}(x)$ tangent to the boundaries of A_i, A_j and A_k. Such a disk is called a *Voronoi tritangent disk*, and more precisely an interior Voronoi tritangent disk if it is included in $A_i \cap A_j \cap A_k$, or an exterior Voronoi tritangent disk if it is lies in the complement of $A_i^\circ \cup A_j^\circ \cup A_k^\circ$.

Suppose we want to add a new object $A \notin \mathcal{A}$ and update the Voronoi diagram from $\mathcal{V}(\mathcal{A})$ to $\mathcal{V}(\mathcal{A}^+)$ where $\mathcal{A}^+ = \mathcal{A} \cup \{A\}$. We assume that $\mathcal{A}^+$ is also an sc-pseudo-circles set. The object A is said to be in conflict with a point x on the 1-skeleton of the current diagram if the Voronoi disk associated to x is either an internal Voronoi disk included in A° or an exterior Voronoi disk intersecting A°. We call *conflict region* the subset of the 1-skeleton of $\mathcal{V}(\mathcal{A})$ that is in conflict with the new object A. A Voronoi edge of $\mathcal{V}(\mathcal{A})$ is said to be in conflict with A if some part of this edge is in conflict with A.

Our dynamic algorithm relies on the two following theorems, which can be proved as in [8].

Theorem 6 *Let $\mathcal{A}^+ = \mathcal{A} \cup \{A\}$ be an sc-pseudo-circles set such that $A \notin \mathcal{A}$. The conflict region of A with respect to $\mathcal{V}(\mathcal{A})$ is a connected subset of the 1-skeleton of $\mathcal{V}(\mathcal{A})$.*

Theorem 7 *Let $\{A_i, A_j, A_k\}$ be an sc-pseudo-circles set in general position. Then the Voronoi diagram of A_i, A_j and A_k has at most two Voronoi vertices.*

This theorem is equivalent to saying that two bisecting curves π_{ij} and π_{ik} relative to the same object A_i have at most two points of intersection. In particular, it implies that the conflict region of a new object A contains at most two connected subsets of each edge of $\mathcal{V}(\mathcal{A})$.

The data structures. The Voronoi diagram $\mathcal{V}(\mathcal{A})$ of the current set of objects is maintained through its dual graph $\mathcal{D}(\mathcal{A})$.

When a deletion is performed, a hidden site can reappear as visible. Therefore, we have to keep track of hidden sites. This is done through an additional data structure that we call the *covering graph* $\mathcal{K}(\mathcal{A})$. For each hidden object A_i, we call *covering set* of A_i a set $K(A_i)$ of objects such that any maximal disk included in A_i is included in the interior of at least one object of $K(A_i)$. In other words, in the Voronoi diagram $\mathcal{V}(K(A_i) \cup \{A_i\})$ the Voronoi cell $V(A_i)$ of A_i is empty. The covering graph is a directed acyclic graph with a node for each object. A node associated to a visible object is a root. The parents of a hidden object A_i are objects that form a covering set of A_i. The parents of a hidden object may be hidden or visible objects.

Note that if we perform only insertions or if it is known in advance that all sites will have non-empty Voronoi cells (e.g., this is the case for disjoint objects), it is not necessary to maintain a covering graph.

The algorithm needs to perform closest site queries. Such a query takes a point x as input and asks for the object in the current set $\mathcal{A}$ that is closest to x. The algorithm maintains a location data structure to perform efficiently those queries. The location data structure that we present here is called a Voronoi hierarchy and is described below in Section 4.

3.1 The insertion procedure

The insertion of a new object A in the current Voronoi diagram $\mathcal{V}(\mathcal{A})$ involves the following steps:

1. Find a first conflict between an edge of $\mathcal{V}(\mathcal{A})$ and A or detect that A is hidden in $\mathcal{A}^+$.
2. Find the whole conflict region of A.
3. Repair the dual graph.
4. Update the covering graph.
5. Update the location data structure if any.

Steps 1, 4 and 5 are discussed below. Steps 2 and 3 are performed exactly as in [6] for the case of disks. Briefly, in Step 2 we perform a depth-first search on the 1-skeleton of $\mathcal{V}(\mathcal{A})$ starting from the first conflict found in Step 1. The boundary points of the conflict region of A with respect to $\mathcal{V}(\mathcal{A})$ are the Voronoi vertices of the Voronoi cell of A in $\mathcal{V}(\mathcal{A}^+)$. Once we have found the conflict region of A , we can construct the Voronoi cell of A in $\mathcal{V}(\mathcal{A}^+)$ by connecting these boundary points in the correct order. In the dual, Step 2 corresponds to finding the boundary of the star of A in $\mathcal{D}(\mathcal{A}^+)$. This boundary represents a hole in $\mathcal{D}(\mathcal{A})$, i.e., a sequence of edges of $\mathcal{D}(\mathcal{A})$ forming a topological circle. Step 3 simply amounts to "staring" this hole from A, that is to connect the vertex in $\mathcal{D}(\mathcal{A}^+)$ associated with A to every vertex on the hole boundary.

Finding the first conflict or detecting a hidden object. The first crucial operation to perform when inserting a new object is to determine if the inserted object is hidden or not. If the object is hidden we need to find a covering set for this object. If the object is not hidden we need to find an edge of the current diagram in conflict with the inserted object.

The detection of the first conflict is based on closest site queries. Such a query takes a point x as input and asks for the object in the current set $\mathcal{A}$ that is closest to x. If we don't have any location data structure, then we perform the following *simple walk* on the Voronoi diagram to find the object in $\mathcal{A}$ closest to x. The walk starts from any object $A_i \in \mathcal{A}$ and compares the distance $\delta(x, A_i)$ with the distances $\delta(x, A)$ to the neighbors A of A_i in the Voronoi diagram $\mathcal{V}(\mathcal{A})$. Here and in the following, two objects are said to be neighbors in the Voronoi diagram if their Voronoi cells are adjacent through an edge. If some neighbor A_j of A_i is found closer to x than A_i, the walk proceeds to A_j. If there is no neighbor of A_i that is closer to x than A_i, then A_i is the object closest to x among all objects in $\mathcal{A}$. It is easy to see that this walk can take linear time. We postpone until the next section the description of the location data structure and the way these queries can be answered more efficiently.

Let us consider first the case of disjoint objects. In this case there are no hidden objects and each object is included in its own cell. We perform a closest site query for any point p of the object A to be inserted. Let A_i be the object of $\mathcal{A}$ closest to p. The cell of A_i will shrink in the Voronoi diagram $\mathcal{V}(\mathcal{A}^+)$ and at least one edge of $\partial V(A_i)$ is in conflict with A. Hence, we only have to look at the edges of $\partial V(A_i)$ until we find one in conflict with A.

When objects do intersect, we perform an operation called *location of the medial axis*, which either provides an edge of $\mathcal{V}(\mathcal{A})$ that is in conflict with A, or returns a covering set of A. To explain

how this operation is performed, let us say that a point p of the medial axis $S(A)$ of A is *covered* by A_i if the disk $M(p)$, which is the maximal disk in A centered at p, is included in the interior A_i° of some object $A_i \in \mathcal{A}$. Let us recall that the medial axis $S(A)$ of A is a tree embedded in the plane and that, from the proof of Theorem 4, we know that the part of $S(A)$ which is not covered by any subset of objects in $\mathcal{A}$ is connected. Roughly, the method consists in pruning recursively the parts of $S(A)$ covered by objects in $\mathcal{A}$ until either: (1) the remaining part of $S(A)$ is empty or (2) we have found a point p of $S(A)$ whose maximal disk $M(p)$ is not covered by the object A_i of $\mathcal{A}$ which is closest to p (if p is not covered by A_i it will not be covered by any other object in $\mathcal{A} \setminus \{A_i\}$). In the first case, A is hidden and the objects in $\mathcal{A}$ which have been used to prune $S(A)$ form a covering of A. In the second case, the point p is in the cell of A_i and at least one edge of $\partial V(A_i)$ is in conflict with A.

Let us explain more precisely, how we select the objects in $\mathcal{A}$ covering parts of $S(A)$ or detect an uncovered point of $S(A)$. We start from a leaf vertex p of the medial axis $S(A)$ and issue a closest site query to find the object A_i closest to p. If the maximal disk $M(p)$ is not covered by A_i we are done. Otherwise we prune the part of $S(A)$ covered by A_i and start again with a new leaf point p', on the boundary of the pruned part of $S(A)$. There is no need to issue a new closest site query for point p' or subsequent considered leaf points on $S(A)$. Indeed because p' is an endpoint of the part of $S(A)$ covered by A_i, the maximal circle $M(p')$ of A centered at p' is also internally tangent to A_i. therefore we just need to scan the neighbors of A_i in the Voronoi diagram $\mathcal{V}(\mathcal{A})$ searching for a neighbor covering $M(p')$. If one is found, it becomes the next object A_j used to prune $S(A)$. If none is found, we know that A_i is closest to p' among all objects in $\mathcal{A}$ and that one of the edge of $\partial V(A_i)$ is in conflict with A.

Updating the covering graph. We now describe how Step 4 of the insertion procedure is performed. We start by creating a node for A in the covering graph.

If A is hidden, the location of its medial axis yields a covering set $K(A)$ of A. In the covering graph we simply assign the objects in $K(A)$ as parents of A.

If the inserted object A is visible, some objects in $\mathcal{A}$ can become hidden due to the insertion of A. The set of objects that become hidden because of A are provided by Step 2 of the insertion procedure. They correspond to cycles in the conflict region of A. The next lemma ensures that the covering graph can be updated by looking at the neighbors of A in the new Voronoi diagram.

Lemma 8 *Let $\mathcal{A}$ be an sc-pseudo-circles set. Let $A \notin \mathcal{A}$ be an object such that $\mathcal{A}^+ = \mathcal{A} \cup \{A\}$ is also an sc-pseudo-circles set and A is visible in $\mathcal{V}(\mathcal{A}^+)$. If an object $A_i \in \mathcal{A}$ becomes hidden upon the insertion of A, then the neighbors of A in $\mathcal{V}(\mathcal{A}^+)$ along with A is a covering set of A_i.*

Proof. Let $N(A_i)$ and $N^\circ(A_i)$ be the parts of the medial axis $S(A_i)$ relative to the diagram $\mathcal{V}(\mathcal{A})$ as defined in Theorem 4. Since A_i becomes hidden in $\mathcal{A}^+$, for every $y \in N^\circ(A_i)$, $M_i(y)$ is contained in A. Let x be a point on the boundary $\partial N(A_i) = N(A_i) \setminus N^\circ(A_i)$ of $N(A_i)$. Then $M_i(x)$ is tangent to some neighbor A_j of A_i in $\mathcal{V}(\mathcal{A})$ and included in A_j. Moreover, we know from the proof of Theorem 4 that if x' belongs to the same connected component of $S(A_i) \setminus N(A_i)$ as x, then $M_i(x')$ is also contained in A_j. Hence the neighbors of A_i in $\mathcal{V}(\mathcal{A})$ along with A form a covering set of A_i. A neighbor A_j of A_i in $\mathcal{V}(\mathcal{A})$ is either a neighbor of A in $\mathcal{V}(\mathcal{A}^+)$ or it becomes hidden in

this diagram. Therefore the set of neighbors of A in $\mathcal{V}(\mathcal{A}^+)$, plus A is a covering set for any object A_i that becomes hidden upon the insertion of A. □

Let A_i be an object that becomes hidden upon the insertion of A. By Lemma 8 the set of neighbors of A in $\mathcal{V}(\mathcal{A}^+)$ along with A is a covering set $K(A_i)$ of A_i. The only modification we have to do in the covering graph is to assign all objects in $K(A_i)$ as parents of A_i.

Updating the location data structure. The location data structure , if any, is updated as follows. Let A be the object inserted. If A is hidden we do nothing. If A is not hidden, we insert A in the location data structure, and delete from it all objects than become hidden because of the insertion of A.

3.2 The deletion procedure

Let A_i be the object to be deleted and let $K_p(A_i)$ be the set of all objects in the covering graph $\mathcal{K}(\mathcal{A})$ that have A_i as parent. The deletion of A_i involves the following steps:

1. Remove A_i from the dual graph.
2. Remove A_i from the covering graph.
3. Remove A_i from location data structure.
4. Reinsert the objects in $K_p(A_i)$.

Step 1 requires no action if A_i is hidden. If A_i is visible, we first build an annex Voronoi diagram for the neighbors of A_i in $\mathcal{V}(\mathcal{A})$ and use this annex Voronoi diagram to fill in the cell of A_i (see [6]). In Step 2, we simply delete all edges of $\mathcal{K}(\mathcal{A})$ to and from A_i, as well as the node corresponding to A_i. In Step 3, we simply delete A_i from the location data structure. Finally, in Step 4 we apply the insertion procedure to all objects in $K_p(A_i)$. Note, that if A_i is hidden, this last step amounts to finding a new covering set for all objects in $K_p(A_i)$.

4 Closest site queries

The location data structure is used to answer closest site queries. A closest site query takes as input a point x and asks for the object in the current set $\mathcal{A}$ that is closest to x. Such queries can be answered through a simple walk in the Voronoi diagram (as described in the previous section) or using a hierarchical data structure called the Voronoi hierarchy.

The Voronoi hierarchy. The hierarchical data structure used here, denoted by $\mathcal{H}(\mathcal{A})$, is inspired from the Delaunay hierarchy proposed by Devillers [5].

The data structure consists in a sequence of Voronoi diagrams $\mathcal{V}(\mathcal{A}_\ell)$, $\ell = 0, \ldots, L$, built for subsets of $\mathcal{A}$ forming a hierarchy, i.e, $\mathcal{A} = \mathcal{A}_0 \supseteq \mathcal{A}_1 \supseteq \ldots \supseteq \mathcal{A}_L$.

The hierarchy $\mathcal{H}(\mathcal{A})$ is built together with the Voronoi diagram $\mathcal{V}(\mathcal{A})$ according to the following rules. Any object of $\mathcal{A}$ is inserted in $\mathcal{V}(\mathcal{A}_0) = \mathcal{V}(\mathcal{A})$. If A has been inserted in $\mathcal{V}(\mathcal{A}_\ell)$ and is visible,

it is inserted in $\mathcal{V}(\mathcal{A}_{\ell+1})$ with probability β. If, upon the insertion of A in $\mathcal{V}(\mathcal{A})$, an object becomes hidden it is deleted from all diagrams $\mathcal{V}(\mathcal{A}_\ell)$, $\ell > 0$, in which it has been inserted. Finally, when an object A_i is deleted from the Voronoi diagram $\mathcal{V}(\mathcal{A})$, we delete A_i from all diagrams $\mathcal{V}(\mathcal{A}_\ell)$, $\ell \geq 0$, in which it has been inserted. Note that all diagrams $\mathcal{V}(\mathcal{A}_\ell)$, $\ell > 0$, do not contain any hidden objects and that if $A \in \mathcal{A}_\ell$, then $A \in \mathcal{A}_{\ell'}$, for all $\ell' < \ell$.

The closest site query for a point x is performed as follows. The query is first performed in the top-most diagram $\mathcal{V}(\mathcal{A}_L)$ using the simple walk. Then, for $\ell = L-1, \ldots, 0$ a simple walk is performed in $\mathcal{V}(A_\ell)$ from $A_{\ell+1}$ to A_ℓ where $A_{\ell+1}$ (resp. A_ℓ) is the object of $\mathcal{A}_{\ell+1}$ (resp. of $\mathcal{A}_\ell$) closest to x.

It easy to show that the expected size of $\mathcal{H}(\mathcal{A})$ is $O(\frac{1}{1-\beta}\, n)$, and that the expected number of levels in $\mathcal{H}(\mathcal{A})$ is $O(\log_{1/\beta} n)$. Moreover, the following lemma, proves that the expected number of steps performed by the walk at each level is constant.

Lemma 9 *Let x be a point in $\mathbb{E}^2$. Let A_ℓ (resp. $A_{\ell+1}$) be the object closest to x in $\mathcal{A}_\ell$ (resp. $\mathcal{A}_{\ell+1}$). Then the expected number of Voronoi cells (objects) visited during the walk in $\mathcal{V}(\mathcal{A}_\ell)$ from $A_{\ell+1}$ to A_ℓ is $O(1/\beta)$.*

Proof. The objects visited at level ℓ are closer to x than $A_{\ell+1}$ and their distances to x are monotonically decreasing. Consequently, if $A_{\ell+1}$ is, among the objects of $\mathcal{A}_\ell$, the k-th closest to x, the walk at level ℓ performs at most k steps. The probability for $A_{\ell+1}$ to be, among the objects of $\mathcal{A}_\ell$, the k-th closest to x is just $\beta(1-\beta)^{k-1}$ and therefore the expected number N_ℓ of objects visited at level ℓ is bounded as follows:

$$N_\ell \leq \sum_{k=1}^{n_\ell} k(1-\beta)^{k-1}\beta < \beta \sum_{k=1}^{\infty} k(1-\beta)^{k-1} = \frac{1}{\beta},$$

where n_ℓ denotes the cardinality of $\mathcal{A}_\ell$. □

We still have to bound the time spent in each one of the visited cells. Let A_i be the site of a visited cell in $\mathcal{V}(\mathcal{A}_\ell)$. Since the complexity of all cells in the Voronoi diagram $V(\mathcal{A}_\ell)$ is only $O(n_\ell)$, where n_ℓ is the number of sites in $\mathcal{A}_\ell$, it is not efficient to simply compare the distances $\delta(x, A_i)$ and $\delta(x, A)$ for each neighbor A of A_i in $\mathcal{V}(\mathcal{A}_\ell)$. This would only imply that the time spent at each level ℓ of the hierarchy is $O(n_\ell) = O(n)$, yielding a total of $O(n)$ time per insertion.

To remedy this we attach an additional balanced binary tree to each cell of each Voronoi diagram in the hierarchy. The tree attached to the cell $V_\ell(A_i)$ of A_i in the diagram $\mathcal{V}(\mathcal{A}_\ell)$ includes, for each Voronoi vertex v of $V_\ell(A_i)$, the ray $\rho_i(p_v)$ where p_v is the point on ∂A_i closest to v, and $\rho_i(p_v)$ is defined as the ray starting from the center of the maximal disk $M_i(p_v)$ and passing through p_v. The rays are sorted according to the (counter-clockwise) order of the points p_v on ∂A_i. When $V_\ell(A_i)$ is visited, the ray $\rho_i(p_x)$ corresponding to the query point x is localized using the tree; here p_x denotes the point of tangency of the disk $C_i(x)$ with the boundary ∂A_i of A_i. Suppose that $\rho_i(p_x)$ is found to be between the rays of two vertices v_1 and v_2. Then it suffices to compare $\delta(x, A_i)$ and $\delta(x, A_j)$, where A_j is the neighbor of A_i in $\mathcal{V}(\mathcal{A}_\ell)$ sharing the vertices v_1 and v_2. Thus the time spend in each visited cell of $\mathcal{V}(\mathcal{A}_\ell)$ is $O(\log n_\ell) = O(\log n)$, which (together with with the expected number of visited nodes) yields the following lemma

Lemma 10 *Using a hierarchy of Voronoi diagrams with additional binary trees for each cell, a closest site query can be answered in time* $O(\frac{1}{\beta \log(1/\beta)} \log^2 n)$.

5 Complexity analysis

In this section we deal with the cost of the basic operations of our dynamic algorithm. We consider three scenarios. The first one assumes that objects do not intersect. In the second scenario objects may intersect but it is assumed that there are no hidden objects. The third scenario is the general case where we allow intersecting and hidden objects.

In each of the above three cases, we consider the expected cost of the basic operations, namely insertion and deletion. The expectation refers to the insertion order, that is, all possible insertion orders are considered to be equally likely and each deletion is considered to deal equally likely with any object in the current set.

Our results are summarized in the table below.

	Disjoint objects	No hidden objects	General case
Insertion	$O(\log^2 n)$	$O(n)$	$O(n)$
Deletion	$O(\log^3 n)$	$O(n)$	$O(n^2)$

Disjoint objects. If the objects are disjoint, there are no hidden objects and therefore there is no need to maintain a covering graph. In this case we use a Voronoi hierarchy as location data structure and therefore we have to take into account the cost for maintaining this structure. Note that the Voronoi hierarchy introduce another source of randomization, which is independent from the randomization in the order of insertion.

Let us first analyze the cost of an insertion.

- Using the Voronoi hierarchy a closest site query can be performed in $O(\log^2 n)$ time. Knowing the object A_i closest to a point p of the inserted object, the tree corresponding to cell $V(A_i)$ in the first level of the hierarchy can be used to find the first edge of $\mathcal{V}(\mathcal{A})$ in conflict with A in time $O(\log n)$.
- Finding the whole conflict region at level ℓ of the hierarchy and updating the Voronoi diagram $\mathcal{V}(\mathcal{A}_\ell)$ can be performed in time $O(k_\ell)$ where k_ℓ is the number of changes occurring in the diagram. The update of additional trees at level ℓ of the hierarchy can be performed in time $O(k_\ell \log n_\ell)$, where n_ℓ is the number of sites in $\mathcal{V}(\mathcal{A}_\ell)$.

At each level ℓ, we have $\log n_\ell = O(\log n)$, and the expected value for k_ℓ is constant. Because the expected number of levels is $O(\log n)$, the expected cost for updating the diagram and the hierarchy upon an insertion is $O(\log^2 n)$.

Let us now analyze the cost of a deletion.

- Updating the Voronoi diagram at level ℓ involves computing a secondary Voronoi diagram involving only the k_ℓ neighbors of the deleted site in the diagram $V(\mathcal{A}_\ell)$. The neighbors are

inserted in random order in the secondary diagram. From what precedes, the expected cost for building this diagram is $O(k_\ell \log^2 n_\ell)$, with the expectation here referring to the random insertion order of the neighbors.

- The cost for updating the additional trees of the location data structure at level ℓ is still $O(k_\ell \log n_\ell)$.

As above, the expected value of k_ℓ, $\ell \geq 0$, is $O(1)$. Moreover, for all levels ℓ, $\log n_\ell = O(\log n)$ and the expected number of levels is $O(\log n)$. Therefore the expected cost for a deletion is $O(\log^3 n)$.

No hidden objects. Assume now that the objects may intersect but that there are no hidden objects. In this case, we maintain neither a covering graph nor a location data structure. Indeed, to find a first conflict we need to perform a location of the medial axis operation. Because this operation has a linear complexity with respect to the number of sites, there is no reason to maintain a location data structure for fast closest site queries.

Then, the analysis of an insertion operation is as above except that finding the first conflict now costs $O(n)$ and, clearly, there is no cost associated with updating the location data structure. Therefore, the expected cost of an insertion is $O(n)$.

The cost of a deletion reduces to the expected cost of building the secondary Voronoi diagram involving the k neighbors of the deleted site. In view of the insertion analysis, it takes $O(k^2)$ expected time to create this secondary Voronoi diagram, with the expectation referring to the random insertion order of the neighbors of the site to be deleted. Taking into account that $k = O(n)$ and that the expected value of k is $O(1)$, we conclude that the expected cost of a deletion is $O(n)$.

General case. In the general case, objects may intersect and/or may be hidden. We maintain a covering graph but no location data structure graph.

Let us analyze the cost of an insertion.

- If the inserted object is hidden, the location of the medial axis provides a covering set for this object, and updating the covering graph has a complexity proportional to the size of the covering set, which is $O(n)$.

- If the inserted object is not hidden, it may induce the hiding of other objects. Objects hidden on an insertion are detected while finding the whole conflict zone. In this case, the analysis of the insertion cost is just as above except that we need to add the cost for updating the conflict graph. This cost is $O(hk')$, where h is the number of objects that become hidden and k' the number of neighbors of the newly inserted object in the updated Voronoi diagram. Because, $h = O(n)$ and k' has a constant expected value, the expected cost to update the conflict graph is $O(n)$.

Thus, in both cases, the expected cost of an insertion is $O(n)$.

Let us come to the analysis of a deletion.

- Because the expected cost of an insertion is linear, the cost of Step 1 (removal of the deleted object from the dual graph) is still $O(k^2)$, where k is the number of neighbors of the deleted site. As above, this gives an expected contribution of $O(n)$.

- Step 2 (removal of the deleted object from the conflict graph) is obviously performed in no more than $O(n)$ time.
- Step 3 induces no cost since we do not maintain a location data structure.
- The cost of Step 4 is $O(nh')$, where h' is the number of sites covered (at least partially) by the deleted site. Obviously, randomization cannot help here to bound the expected number of neighbors of these covered sites and the cost of Step 4 can only be bounded by $O(n^2)$.

Hence, the overall expected cost of a deletion is $O(n^2)$.

6 Extensions

In this section we consider several extensions of the problem discussed in the preceding sections.

Degenerate configurations. Degenerate configurations occur when the set contains pairs of internally tangent objects (cf. Figs. 1(f) and 1(g)). Let $\{A_i, A_j\}$ be an sc-pseudo-circles set with A_i and A_j internally tangent and $A_i \subseteq A_j$. The bisector π_{ij} is homeomorphic to a ray, if A_i and A_j have a single tangent point, or two, in general disconnected, rays, if A_i and A_j have two tangent points. In any case, the interior $V^\circ(A_i)$ of the Voronoi region of A_i in $\mathcal{V}(\{A_i, A_j\})$ is empty and we consider the object A_i as hidden. This point of view is consistent with the definition we gave for hidden sites, which is that an object A is hidden if $N^\circ(A) = \emptyset$.

Let us discuss the algorithmic consequences of allowing degenerate configurations. When the object A is inserted in the diagram, the case where A is internally tangent to a visible object $A_i \in \mathcal{A}$ is detected at Step 1, during the location the medial axis of A. The case of an object $A_j \in \mathcal{A}$ is internally tangent to A is detected during Step 2, when the entire conflict region is searched. In the first case A is hidden and its covering set is $\{A_i\}$. In the second case A_i becomes hidden and its covering set is $\{A\}$. The complexity of insertions and deletions is not affected by allowing these degenerate configurations.

Pseudo-circles sets of piecewise smooth convex objects. In the sections above we assumed that all convex objects have smooth boundaries, i.e., their boundaries are at least C^1-continuous. In fact we can handle quite easily the case of objects whose boundaries are only piecewise C^1-continuous. Let us call *vertices* the points on the boundary of an object where there is no C^1-continuity. The main problem of piecewise C^1-continuous objects is that they can yield two-dimensional bisectors when two objects share the same vertex (cf. Figs. 1(i) and 1(j)). The remedy is similar to the commonly used approach for the Voronoi diagram of segments (e.g., cf. [4]): we consider the vertices on the boundary of the objects as objects by themselves and slightly change the distance so that a point whose closest point on object A_i is a vertex of A_i is considered to be closer to that vertex. All two-dimensional bisectors, if any, then become the Voronoi cells of these vertices.

As far as our basic operations are concerned, we proceed as follows. Let A be the object to be inserted or deleted. We denote by A_v the set of vertices of A and $\hat{A}$ the object A minus the points in A_v. When we want to insert A in the current Voronoi diagram we at first insert all points in A_v and then $\hat{A}$. When we want to delete A we at first delete $\hat{A}$ and then all points in A_v. During the latter

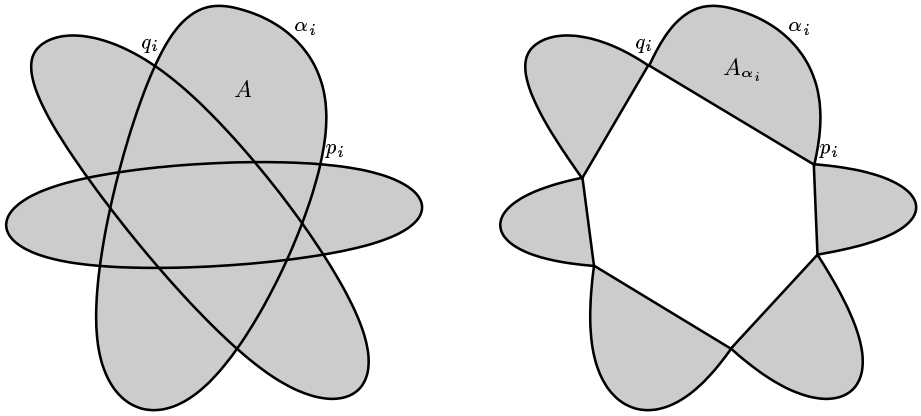

Figure 4: The sets $\mathcal{A}$ (left) and $\mathcal{A}'$ (right). $\mathcal{A}'$ is a generalized pseudo-circles set.

step we have to make sure that points in A_v are not vertices of other objects as well. This can be done easily by looking at the neighbors in the Voronoi diagram of each point in A_v.

Generic convex objects. In the case of smooth convex objects which do not form pseudo-circles sets we can compute the Voronoi diagram in the complement of their union (free space). The basic idea is that the Voronoi diagram in free space depends only on the arcs appearing on the boundary of the union of the objects.

More precisely, let $\mathcal{A}$ be a set of convex objects and let C be a connected component of the union of the objects in $\mathcal{A}$. Along the boundary ∂C of C, there exists a sequence of points $\{p_1, \ldots, p_m\}$, which are points of intersection of objects in $\mathcal{A}$. An arc α_i on ∂C joining p_i to p_{i+1} belongs to a single object $A \in \mathcal{A}$. We form the piecewise smooth convex object A_{α_i}, whose boundary is $\alpha_i \cup p_i p_{i+1}$, where $p_i p_{i+1}$ is the segment joining the points p_i and p_{i+1}. Consider the set $\mathcal{A}'$ consisting of all such objects A_{α_i}. $\mathcal{A}'$ is a pseudo-circles set (consisting of disjoint piecewise smooth convex objects) and the Voronoi diagrams $\mathcal{V}(\mathcal{A})$ and $\mathcal{V}(\mathcal{A}')$ coincide in free space.

The set $\mathcal{A}'$ can be computed by performing a line-sweep on the set $\mathcal{A}$ and keeping track of the boundary of the connected components of the union of the objects in $\mathcal{A}$. This can be done in time $O(n \log n + k)$, where $k = O(n^2)$ is the complexity of the boundary of the afore-mentioned union. Since the objects in $\mathcal{A}'$ are disjoint, we can then compute the Voronoi diagram in free space in total expected time $O((n + k) \log^2 n)$.

7 Conclusion

We presented a dynamic algorithm for the construction of the euclidean Voronoi diagram in the plane for various classes of convex objects. In particular, we considered pseudo-circles sets of piecewise smooth convex objects, as well as generic smooth convex objects, in which case we can compute the

Voronoi diagram in free space. Our algorithm uses fairly simple data structures and enables us to perform deletions easily.

We are currently working on extending the above results to non-convex objects, as well as understanding the relationship between the euclidean Voronoi diagram of such objects and abstract Voronoi diagrams. We conjecture that, given a pseudo-circles set (of possibly non-convex objects) in general position, such that any pair of objects has exactly two supporting lines, the corresponding set of bisectors is an admissible system of bisectors.

Acknowledgments

Work partially supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project IST-2000-26473 (ECG - Effective Computational Geometry for Curves and Surfaces).

References

[1] H. Alt and O. Schwarzkopf. The Voronoi diagram of curved objects. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 89–97, 1995.

[2] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991.

[3] F. Aurenhammer and R. Klein. Voronoi diagrams. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 201–290. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[4] C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections*. Ph.D thesis, Universität des Saarlandes, March 1996.

[5] O. Devillers. The Delaunay hierarchy. *Internat. J. Found. Comput. Sci.*, 13:163–180, 2002.

[6] M. I. Karavelas and M. Yvinec. Dynamic additively weighted Voronoi diagrams in 2D. In *Proc. 10th Europ. Sympos. Alg.*, pages 586–598, 2002.

[7] R. Klein. *Concrete and Abstract Voronoi Diagrams*, volume 400 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1989.

[8] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom.: Theory & Appl.*, 3(3):157–184, 1993.

[9] V. Koltun and M. Sharir. Polyhedral Voronoi diagrams of polyhedra in three dimensions. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 227–236, 2002.

[10] V. Koltun and M. Sharir. Three dimensional euclidean Voronoi diagrams of lines with a fixed number of orientations. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 217–226, 2002.

[11] M. McAllister, D. Kirkpatrick, and J. Snoeyink. A compact piecewise-linear Voronoi diagram for convex sites i n the plane. *Discrete Comput. Geom.*, 15:73–105, 1996.

[12] A. Okabe, B. Boots, K. Sugihara, and S.-N. Chiu. *Spatial tessellations: concepts and applications of Voronoĭ diagrams.* John Wiley & Sons Ltd., Chichester, 2nd edition, 2000.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399